UNITED STATES DISTRICT COURT
SOUTHERN DISTRICT OF NEW YORK

ARISTA RECORDS LLC; ATLANTIC RECORDING CORPORATION; BMG MUSIC; CAPITOL RECORDS, INC.; ELEKTRA ENTERTAINMENT GROUP INC.; INTERSCOPE RECORDS; LAFACE RECORDS LLC; MOTOWN RECORD COMPANY, L.P.; PRIORITY RECORDS LLC; SONY BMG MUSIC ENTERTAINMENT; UMG RECORDINGS, INC.; VIRGIN RECORDS AMERICA, INC.; and WARNER BROS. RECORDS INC., Plaintiffs, v. LIME WIRE LLC; LIME GROUP LLC; MARK GORTON; GREG BILDSON; and M.J.G. LIME WIRE FAMILY LIMITED PARTNERSHIP, Defendants.	Case No. 06 CV 5936 (KMW) ECF Case **DECLARATION OF DR. STEVEN D. GRIBBLE** **SUBJECT TO PROTECTIVE ORDER FILED UNDER SEAL**

I, Dr. Steven D. Gribble, the undersigned, hereby declare:

1. I am over twenty-one (21) years of age and am of sound mind. I have never been convicted of a felony of a misdemeanor involving moral turpitude. I have personal knowledge of the facts stated herein, I am competent to testify thereto, and if called to testify, I could and would testify to the following.

2. I am an Associate Professor in the Department of Computer Science and Engineering at the University of Washington. I received my masters and Ph.D. degrees in Computer Science from the University of California at Berkeley, and my B.Sc. degree in Computer Science and Physics from the University of British Columbia. My teaching and research in part focuses on peer-to-peer systems, content delivery systems, and the Web. More broadly, my research specializes in computer operating systems, distributed systems, and computer security, and I have taught both undergraduate and graduate courses on these topics.

My up-to-date curriculum vitae is included as Exhibit A; my CV also includes a list of my research publications.

3. I have been retained by the Lime Wire to provide expert analysis and opinions on technical issues relating to this litigation. On July 18th, 2008, I submitted a prior declaration in support of Defendants' motions for summary judgment on this matter. For the purpose of this current declaration, Lime Wire's counsel has asked me to review and comment on Professor Horowitz's declaration in support of Plaintiff's motion for a permanent injunction.

Summary of opinions

4. **Hash-based filtering is an effective and precise mechanism that can serve an important role in many filtering system designs.** A hash-based fingerprint is extremely effective at identifying a specific file. Professor Horowitz does identify a technical disadvantage of a hash-based fingerprint, namely that different encodings or variants of a work will have different hash fingerprints, but a well-designed filtering system can exploit the strengths of hash-based filtering while avoiding this weakness. Specifically, in one design, a filtering system can use other, expensive mechanisms to determine whether a file contains potentially infringing content, and then use a hash-based fingerprint to identify instances of that file inexpensively and accurately in the future.

5. **The LimeWire software is not Gnutella.** The LimeWire software is one of many different software packages that people can choose to install and use to interact with the Gnutella peer-to-peer network. A filtering system integrated with the LimeWire software might be able to deter potentially infringing activities involving the LimeWire software itself, but Lime Wire has no way to directly affect potentially infringing activities of people that use programs other than the LimeWire software. In many ways, this situation is similar to how there are many Web browsers that people can choose to use to interact with the Web: even if one Web browser vendor integrates technical mechanisms to try to prevent its browser from being used to access

potentially infringing content on the Web, those mechanisms will not be able to affect the potentially infringing activities of people that use other browsers.

Detailed comments on Professor Horowitz's Declaration

6. I have read Professor Horowitz's declaration and would like to comment on several issues that it raises. Some of my comments are intended to provide the court with additional context on filtering systems, while others will directly concern specific conclusions and opinions that Professor Horowitz provided.

7. **No filtering system will be perfect.** In general, a filtering system faces many tradeoffs and compromises, including:

a. a fundamental tension between minimizing false positives (accidentally filtering out benign files) and minimizing false negatives (accidentally permitting infringing files);

b. balancing the conflicting interests of law-abiding users, infringing and potentially adversarial users seeking to evade filters, the content owners, and software vendors;

c. deciding whether to preserve the architectural advantages of a decentralized P2P network, such as its scalability, fault tolerance stemming from a lack of a single point of failure, and the inability of a single person or organization to control the network, or deciding whether to add centralized or non-P2P elements to the system to better monitor or control user activities, possibly eroding the advantages of decentralization;

d. minimizing the operational costs to software vendors and content owners, while simultaneously minimizing the impact on users with respect to the performance and usability of their software and privacy of their activities on the network.

8. Building an effective filtering system in a peer-to-peer setting is a task that faces significant engineering, if not research, challenges. There is no agreed-upon technical standard

for how this should be done and there are many designs one could consider. Deciding between them involves both obvious and subtle tradeoffs involving the issues I mention above. Given this, building an effective filtering system will likely require iteration and adaptation over time as experience is gained, and it is unreasonable to expect that a "perfect" filtering system can be immediately designed and deployed.

9. **"Hashing," when used properly, is an effective tool that can play a valuable role in a filtering system.** A hash identifies a specific file precisely. As well, a hash is inexpensive and simple to generate, and it also is efficient to store or transmit. Hashes are very useful when two computers across a network want to compare files without having to transmit them to each other: their hashes can be compared instead. As well, if one were to build a database of hashes that correspond to known-to-be-infringing files, a given file can be efficiently tested for inclusion against that set by looking up its hash in the database.

10. The primary disadvantage of hashes is a direct consequence of their precision. If any part of a file changes, then its hash will change as well. A hash cannot identify "semantically related" files, such as two files that contain the same piece of music but were encoded using different software or encoding parameters. As such, hashing is not a useful mechanism for examining a new file that has never been encountered before to determine whether it contains infringing content. However, hashing is extremely useful for comparing a file against a set of previously encountered, known-to-be-infringing files.

11. Professor Horowitz correctly identifies this technical limitation of hashing, but from this specific fact he extrapolates general conclusions that are unsupported and perhaps incorrect, specifically that a filtering system that incorporates hash-based filtering would be ineffective and overly burdensome to the Plaintiffs.

12. In the context of a file-sharing network, an effective filtering system must consist of several components, including the following:

a. An authoritative list of content (e.g., music works) that should be filtered. Without this, a filtering system would not be able to perform its job, since it would have no ground truth against to compare files. This authoritative list must contain enough information that it is possible to generate effective filtering criteria. The information in the authoritative list might include metadata about the content (such as a song title), and better, it could include the content itself (such as an encoding of the song).

b. Some mechanism for comparing a file against this authoritative list of content, to determine whether or not it matches an item in the list. One potential mechanism is to try to match textual elements contained in the file metadata (such as its title, author, or album name, in the case of music) against information in the authoritative list. Textual filtering tends to be imprecise, as it can suffer from both false positives and false negatives, but it is relatively inexpensive to perform. Another potential mechanism is to use acoustic fingerprinting, which performs an algorithmic analysis of audio content to extract enough "features" so that the content can be precisely and uniquely identified. Acoustic fingerprinting is reportedly accurate, though I have not seen a detailed, objective analysis of its false positive and false negative rates over a corpus the size of that corresponding to peer-to-peer networks. It is my understanding that acoustic fingerprints are more expensive to generate, store, transmit, and compare than hash values or text filters.

13. If a filtering system cannot compute or compare acoustic fingerprints at a high rate, such as the rate at which file transfers are attempted via the LimeWire software, it might instead use a hybrid approach that combines acoustic fingerprinting with more efficient hash comparisons. Once a file has been found to be infringing using the more expensive acoustic fingerprinting, then a hash can be generated for the file and the hash added to a database of known-to-be-infringing files. Hashing can then be used to identify that file in the future quickly, inexpensively, and precisely.

14. For such a hybrid approach to work, the filtering system must decide when, and over which files, acoustic fingerprints should be generated. One approach might be to have a "crawler" continuously probe the Gnutella network for potentially infringing files, such as files that match a loose text filter, and then apply acoustic fingerprinting over the matching subset to precisely determine which are infringing. Any files that are found to be infringing could then be automatically added to a hash-based filter set. Hypothetically, the Plaintiffs, Lime Wire, or third parties could operate this crawler.

15. An alternative approach to a hybrid system would have the LimeWire software generate acoustic fingerprints in some scenarios. For example, LimeWire software involved in a transfer could first check the hash of the file against a list of known-to-be-infringing files. If something matches, the transfer would be denied. If not, the LimeWire software could then generate an acoustic fingerprint of the file, and transmit that to a server for comparison against the authoritative list. If something matches, the transfer would be denied and the hash could be added to the hash database. If not, then the transfer would be allowed. The disadvantages to this scheme are (a) that acoustic fingerprinting code would need to be added to the LimeWire software, and (b) all transfers of non-infringing audio files would cause acoustic fingerprints to be generated and sent to a server for comparison, potentially deluging that server. (You could imagine caching the results of this as well, by creating and maintaining a database of hashes of non-infringing files, but this raises other technical issues.)

16. Many other hybrid designs are possible beyond the two that I outlined above.

17. Diving a little deeper, published measurement studies[1] of content available on and transferred over various peer-to-peer file systems, including Gnutella, have shown that content popularity is highly skewed. In a nutshell, a relatively small fraction of specific, popular files are

[1] See, for example, "Availability and Popularity Measurements of Peer-to-Peer File Systems" by Chu, Labonte, and Levin from the University of Massachusetts, Amherst, and the various papers it cites, or more recently, "Power-law Revisited: A Large Scale Measurement Study of P2P Content Popularity" from Dan and Carlsson of KTH, Royal Institute of Technology, and the University of Calgary.

responsible for large fraction of copies or transfers on these systems; this set of popular files is known as the "head" of the file popularity distribution. In contrast, the "tail" of the distribution consists of many unpopular files, most of which are encountered or transferred only once over the measurement period.

18. Given this, a hybrid filtering system that focuses its effort on identifying and acoustically fingerprinting files within the "head" of the popularity distribution might be adequately effective, and it would likely require less effort to maintain than Professor Horowitz implies. It is admittedly true that new infringing works, and new encodings of old infringing works, might continually appear on the system, and that there associated files would need to be identified and hash fingerprints for them computed and included in the filtering system. But, if these new files become popular, a strategy that focuses on the "head" of the popularity curve would find and include them when they become important enough.

19. From reading Professor Horowitz's declaration, one might form the impression that the use of hashing in a filtering scheme necessarily implies that Plaintiffs would need to expend effort to identify all versions of all files on the Gnutella network on a near constant basis, and that doing anything less would prove ineffective. These studies suggest that this is not true. As well, if Lime Wire operates the "crawler" in the first hybrid filtering scheme I discussed above, then the Plaintiffs would not need to expend any ongoing effort at all.

20. I should emphasize that the filtering designs that I have discussed, and many that the Plaintiffs have proposed or discussed, are hypothetical and have not been deployed or evaluated at scale in a decentralized, peer-to-peer setting such as Gnutella. As such, how effectively they would perform and the issues they would face in practice are speculative, especially over the long term as benign users or adversaries adapt to the system.

21. **The LimeWire software is not Gnutella.** Given this discussion of filtering, it is worth reiterating the role that the LimeWire software plays in the overall Gnutella ecosystem. Gnutella is an open, decentralized peer-to-peer network. Users can interact with Gnutella using

any of several available Gnutella-compatible client programs, such as LimeWire, Gnucleus, or Phex. No one company or software vendor controls the network; as such, a filtering system integrated into the LimeWire software would not directly affect the operation of other clients, nor would it be able to monitor or control all files or transfers performed over Gnutella.

22. In several ways, the technical relationship of the LimeWire software to Gnutella has similarities to the technical relationship between a Web browser such as Internet Explorer (IE) and the Web. There are many different Web browsers available to users, including IE, Safari, Opera, Firefox, and Chrome, some of which are more widely used than others. There are infringing files available through the Web. Even if one were to deploy filtering technologies in a particular Web browser such as IE, this would not prevent infringing files from being made available from some servers and accessed via the other browsers.

I HEREBY DECLARE and certify under penalty of perjury under the laws of the United States of America that the foregoing is true and correct, and that this Declaration was executed this 20th day of June 2010 in Seattle, Washington, USA.